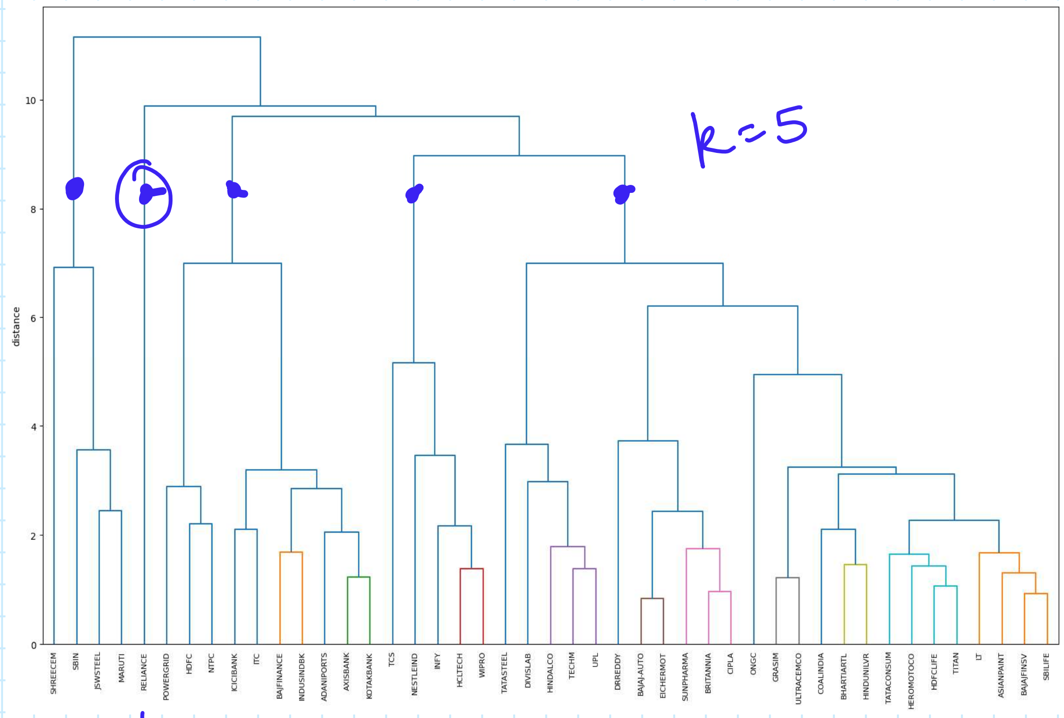
GMM

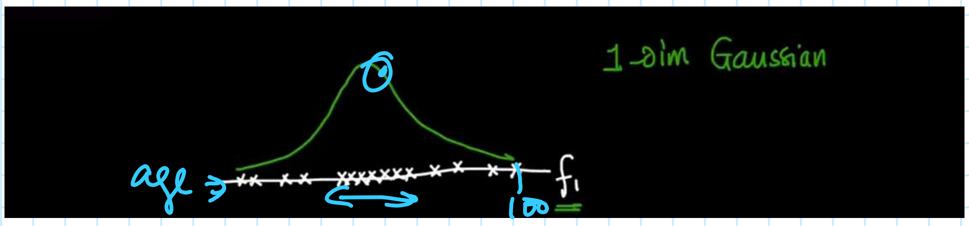
15 February 2024 20:41

Gaussian Mixture Models



outlier

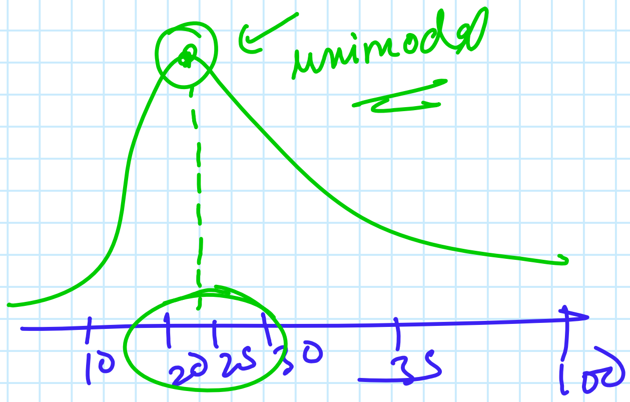
Gaussian



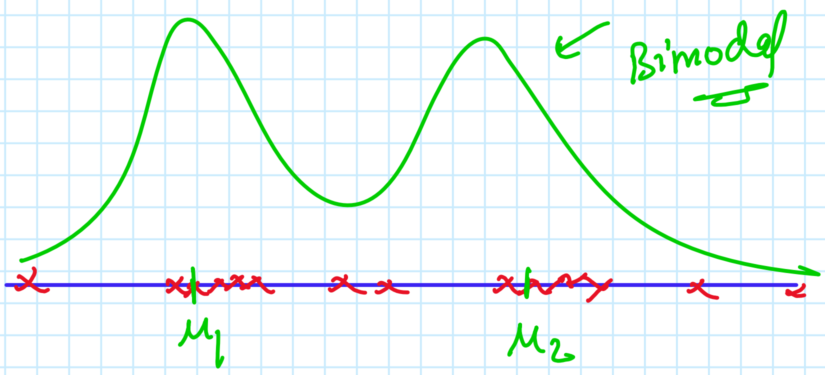
120
20

Modelling

Age
10
20
25
30
35
100

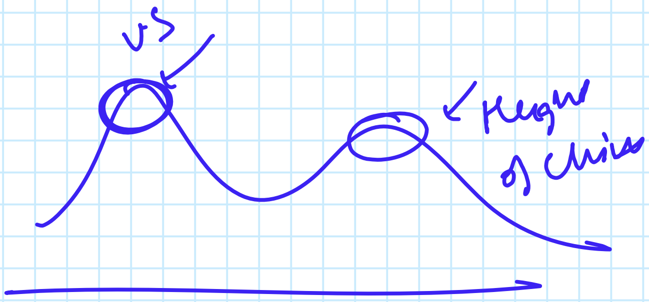


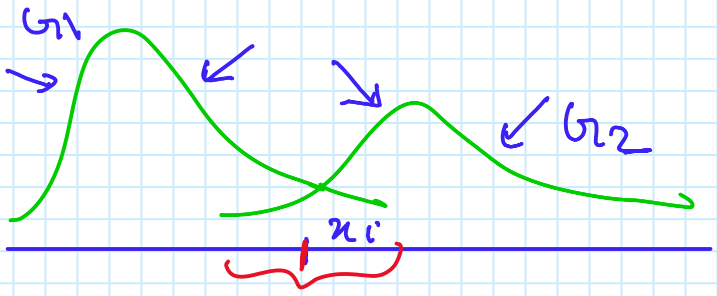
- In that case, there will be two **Gaussian Distributions** with mean μ_1 and μ_2.
- If there is only a single **mount(peak)** in the distribution of a data, then the data is known as **uni-modal data**
- If there are more than one **mount(peak)** in the distribution of a data, then the data is known as **multi-modal data**



If we combine both the gaussians, the point at which both distributions intersects will have more height than the start and the end of the distribution.

This is known as **Mixture of Gaussians**





Q. What does the intersection represents?

- It says that the point at intersection can either belong to distribution G1 or distribution G2

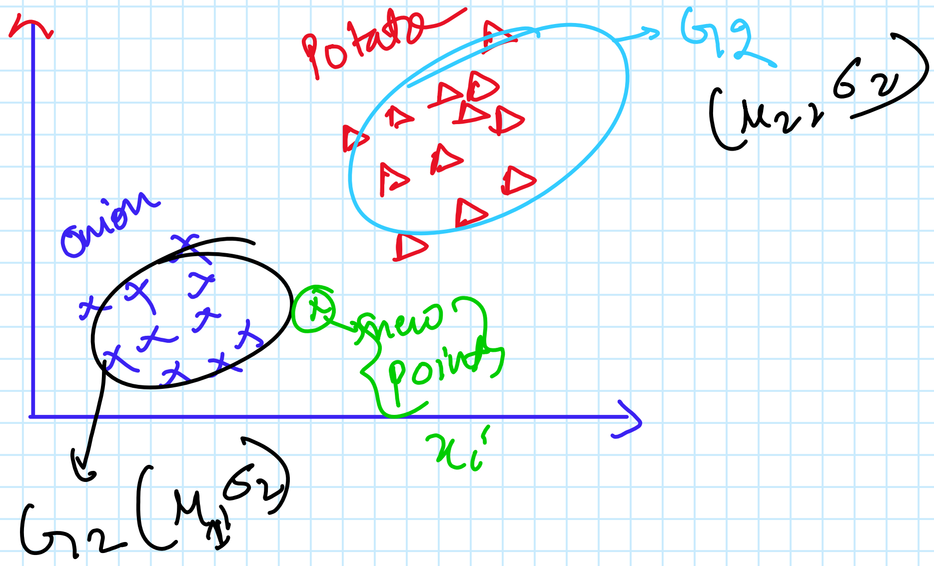
$P(G_1 = x_i) = 0.02$ and $P(G_2 = x_i) = 0.06$

Normalise $\rightarrow$ $\frac{0.02}{0.02+0.06} = 20\%$ G1 $\qquad \frac{0.06}{0.02+0.06} = 80\%$ G2

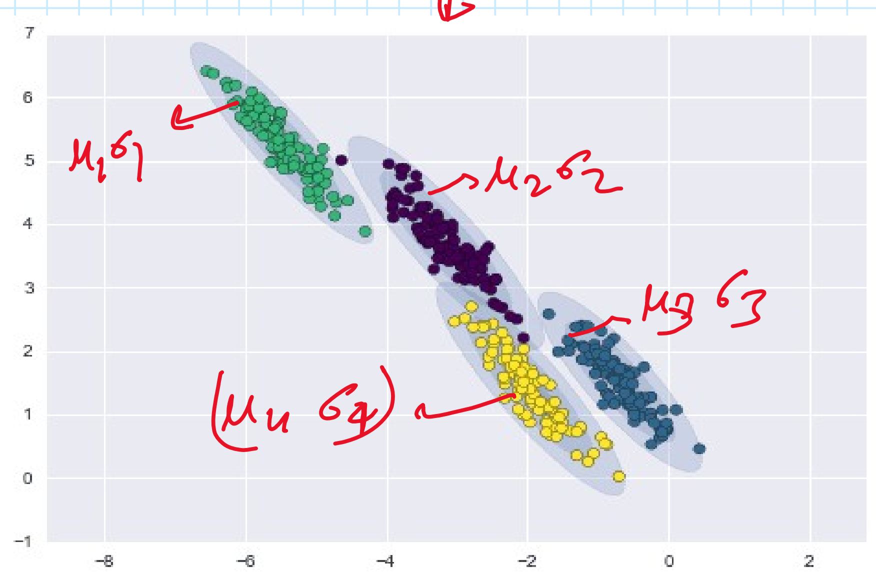
GMM

15 February 2024 21:19

Gaussian Mixture Models → model data as mixture of gaussians



k=4



K → means → Hard Clustering

onion

Potato

$$P(x_i = G_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-z^2/2}, \quad z^2 = \left(\frac{x-\mu_1}{\sigma_1}\right)^2$$

$$P(x_i = G_2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2}$$

⇒ 0.03

⇒ 0.4

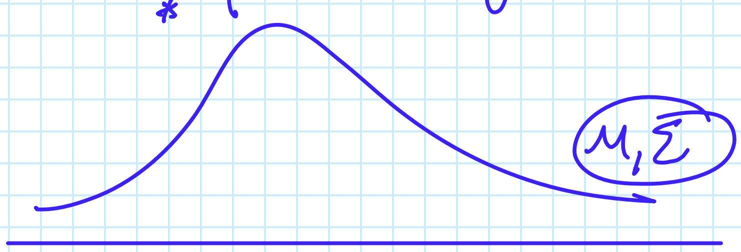
normalise

$$\frac{0.03}{Total = (0.03+0.4)} = 3\%$$

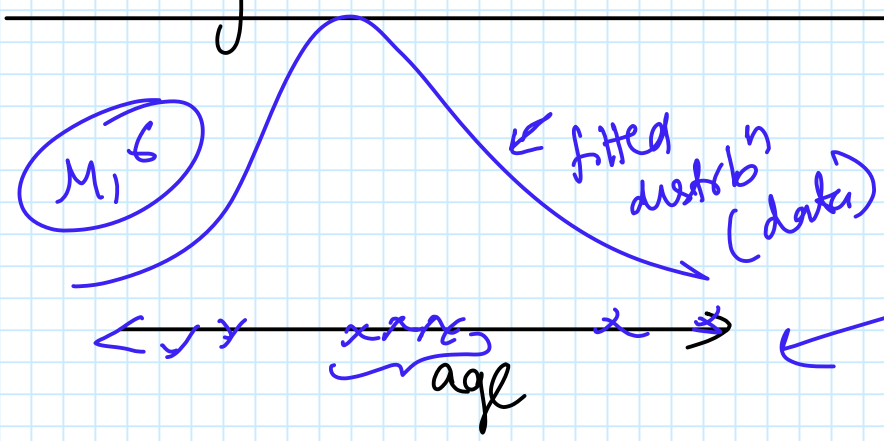
$$\frac{0.4}{Total(0.03+0.4)} = 97\%$$

Gaussian → [completely characterised by Mean & Std Dev]

*



Extending Gaussian to multi Dimensions (2D)

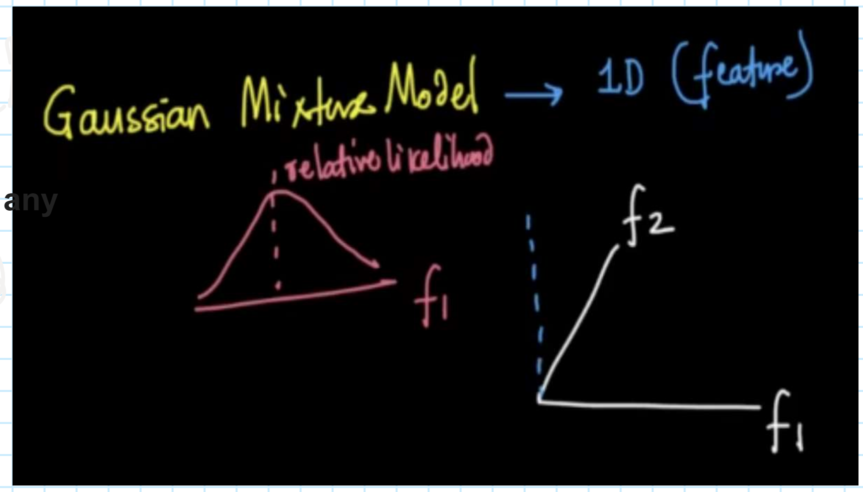


f ⇒

Age
20
25
35
40

$$\frac{20+25+35+40}{4}$$

$$\sqrt{\frac{\Sigma(x_i-\bar{x})^2}{N}}$$



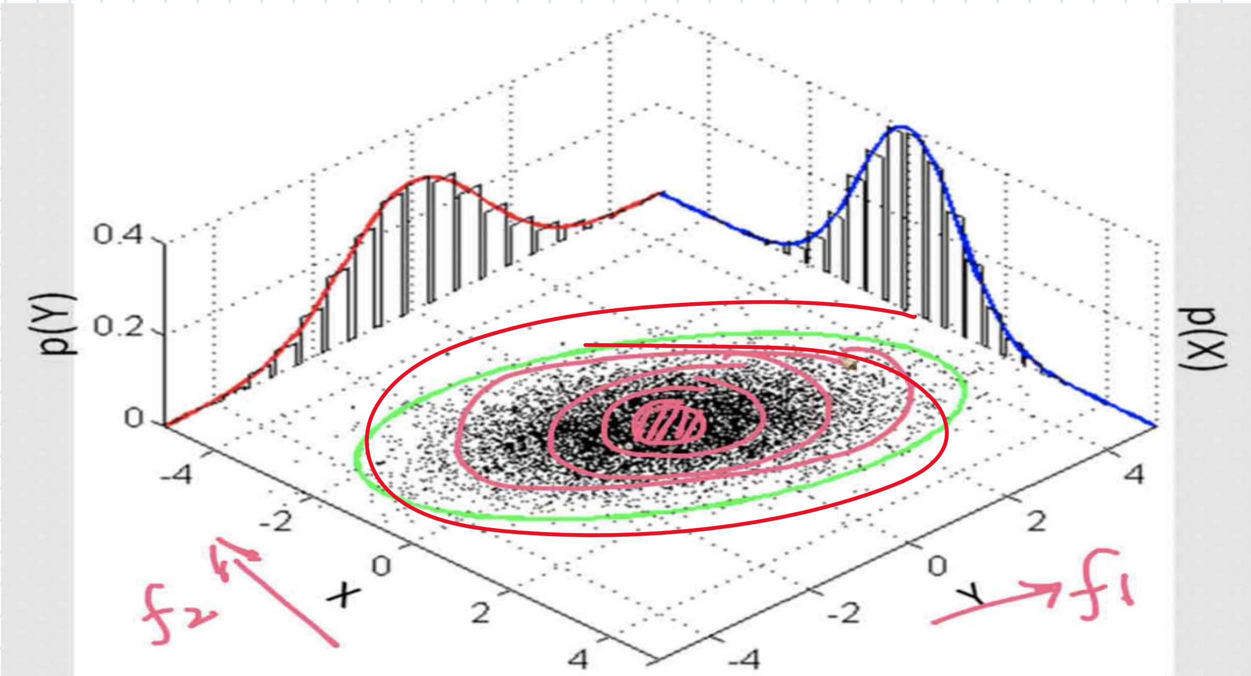
Intuition behind Gaussian Mixture Model

- The basic intuition behind Gaussian Mixture Model is that **we can model any data, that we have, as a mixture of Gaussians**.

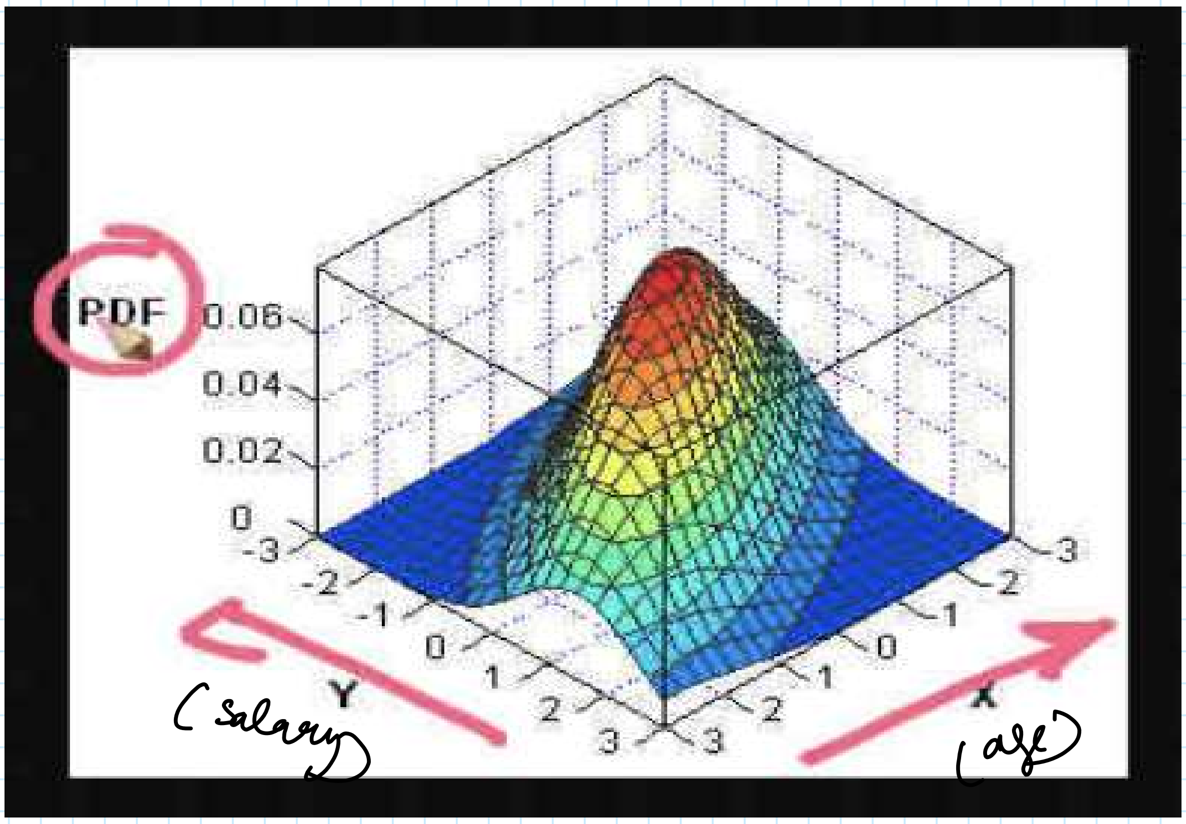
GAUSSIAN IN 2 Dimensions

- As we can see, the data points are more dense in the centre and as we move away from the centre, the density of points decreases. This is to simulate a Gaussian (Normal) Distribution.
- If we consider only 1 feature at a time, we can see its Gaussian Distribution:
  - Red curve represents the Gaussian Distribution for Feature 1 alone.
  - Blue curve represents the Gaussian Distribution for Feature 2 alone.
- The Gaussian Distribution of the 2 features combined would look something like shown below:

* [Gaussian Distribⁿ for Two features]



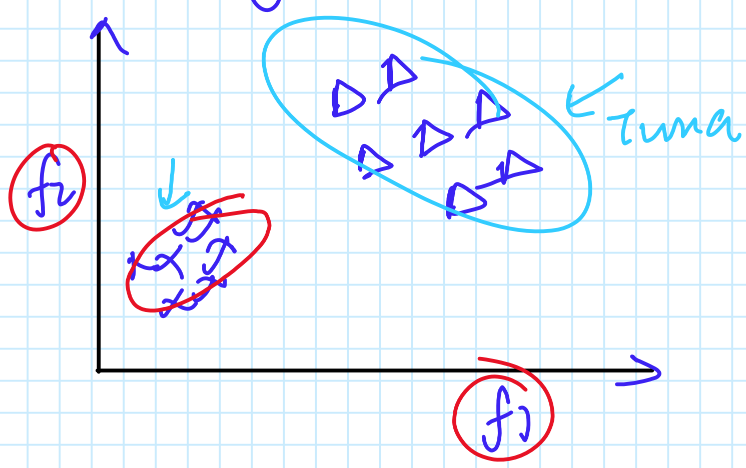
f1 (age)	f2 (salary)



Parameters (2D)

$\vec{\mu}$, Σ variance f1

$\begin{bmatrix}\mu_{age} \\ \mu_{salary}\end{bmatrix}$ $\begin{bmatrix}\sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy}\end{bmatrix}$

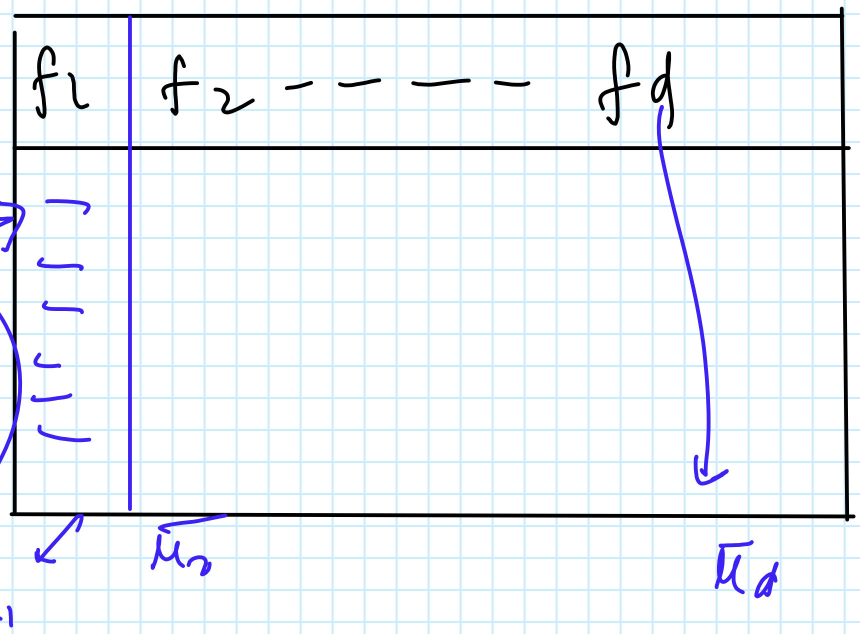


μ, σ → μ Potato, μ onion Σ

GMM

15 February 2024 21:53

D dimension →



$N \sim (\mu^d, \Sigma^{d \times d})$

covariance matrix

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_1^2 & & ? \\ ? & \sigma_2^2 & \\ & & \ddots \\ & & & \sigma_d^2 \end{bmatrix}$$

k=2

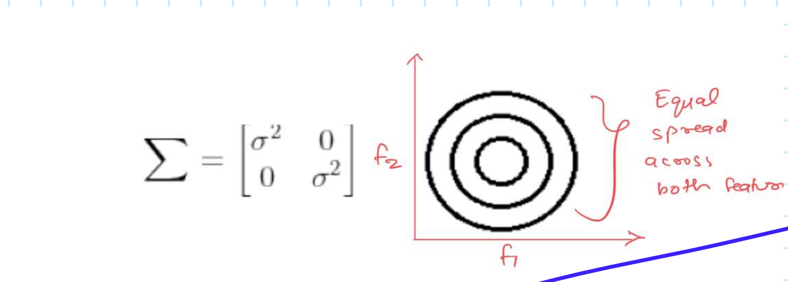
potato

$\begin{bmatrix} \mu_{p f_1} \\ \mu_{p f_2} \end{bmatrix}$ Σ_{pot}

2 Dim Gaussians w k=2

$\begin{bmatrix} \mu_{\text{onion} f_1} \\ \mu_{\text{onion} f_2} \end{bmatrix}$ Σ_{onion}

Ques

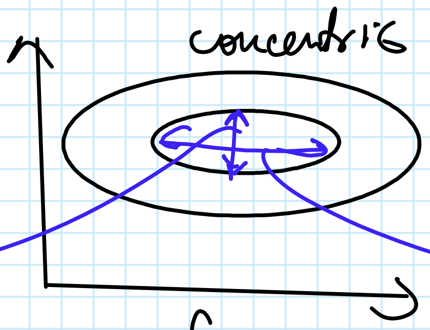


Covariance matrix with equal variance (Image by Author)

II $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$

$(f_1 \perp f_2)$

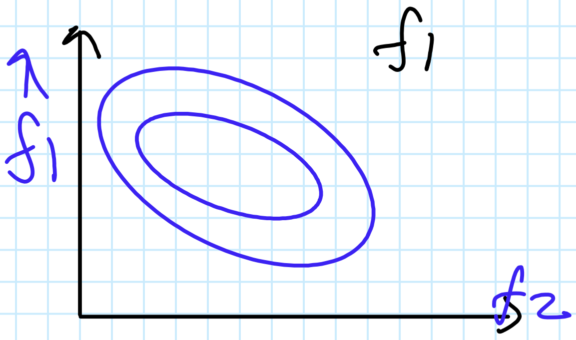
concentric



var(f_1)

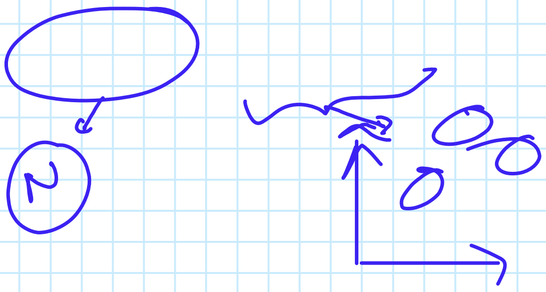
III $\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} =$

$f_1 \not\perp f_2$



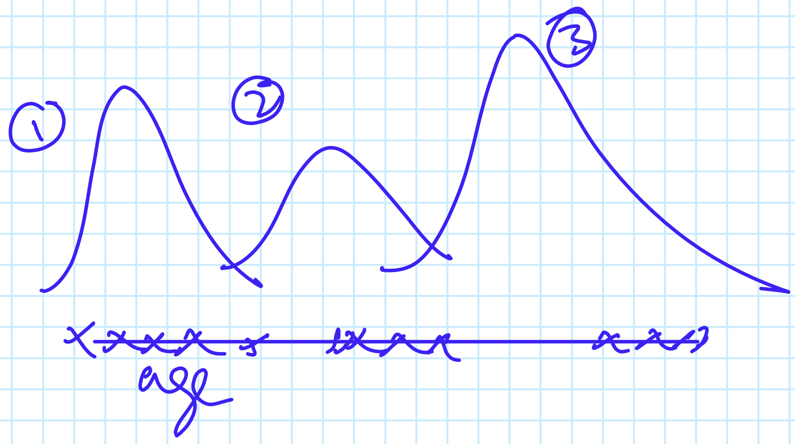
GMM

15 February 2024 22:01

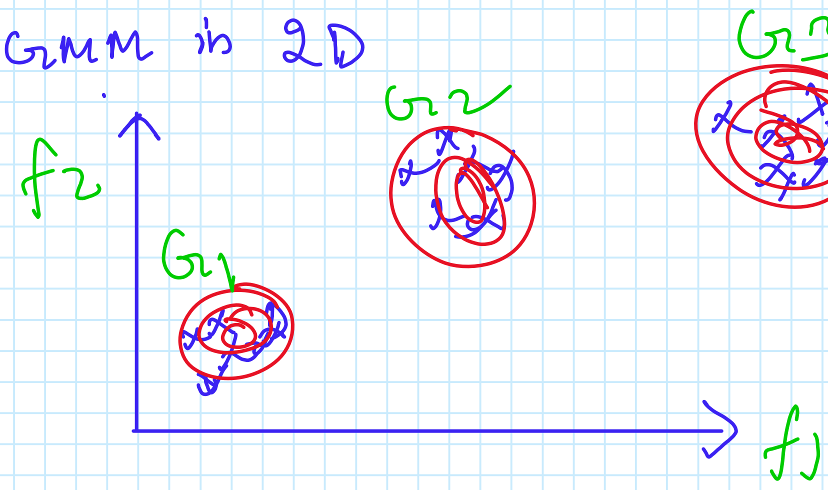


Gaussian Mixture Models

→ GMM in 1D (k=3)



→ GMM in 2D



GAUSSIAN IN D DIMENSION SPACE

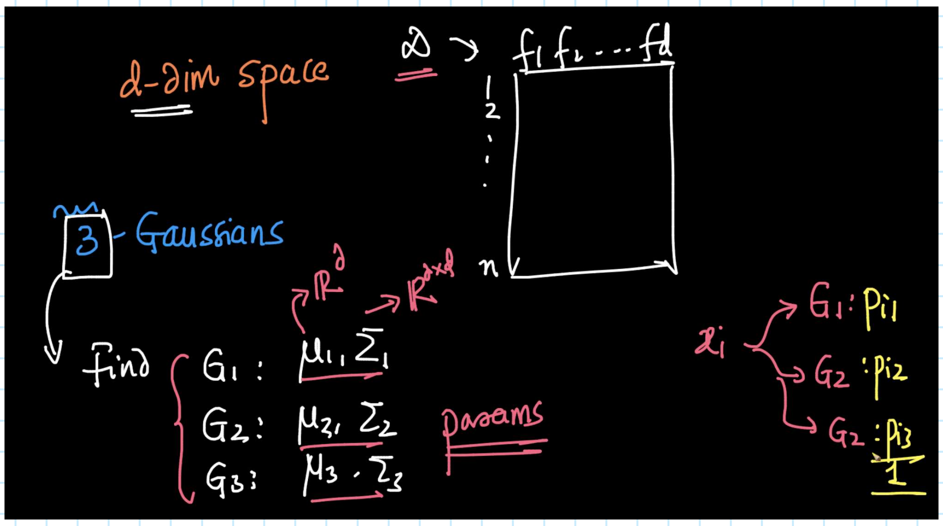
(D → dimensional space)

Each gaussian will have its own set of parameters

$G_1 : \mu_1 \Sigma_1$

$G_2 : \mu_2 \Sigma_2$

$G_3 : \mu_3 \Sigma_2$



x_i → G_1 (0.2) → 0.2/0.8 %

x_i → G_2 (0.1) → 0.1/0.8 %

x_i → G_3 (0.5) → 0.5/0.8 %

For a d-Dimensional Gaussian Mixture Model, we'll need:

- The d-dimensional vector of means μ(d) and the d×d Covariance Matrix ∑(d×d), i.e., the Normal Distribution Nd.
- If we have this Normal Distribution Nd, we can compute the probability of a data point belonging to a Gaussian (Cluster) using a slightly more complex density function:

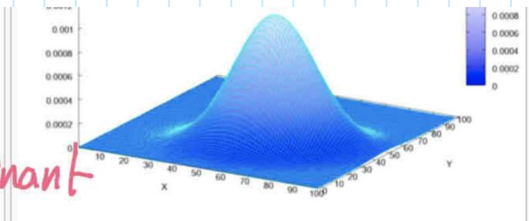
*(Ignore)

$P(x_i \to G_1) =$ (d dim) $\mu_d, \Sigma_{d\times d}$

definite. In this case the distribution has density[5]

$$f_{\mathbf{X}}(x_1,\ldots,x_d) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}$$

√determinant



(PDF of

1 D formula → $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)\left(\frac{x-\mu}{\sigma}\right)}$ $\sqrt{2\pi\sigma^2}$

GMM

15 February 2024 22:26

Training Algo → (Expectation Maximisation)

EM find the parameters of model

K Means → hard → Random initialis" (centroid)

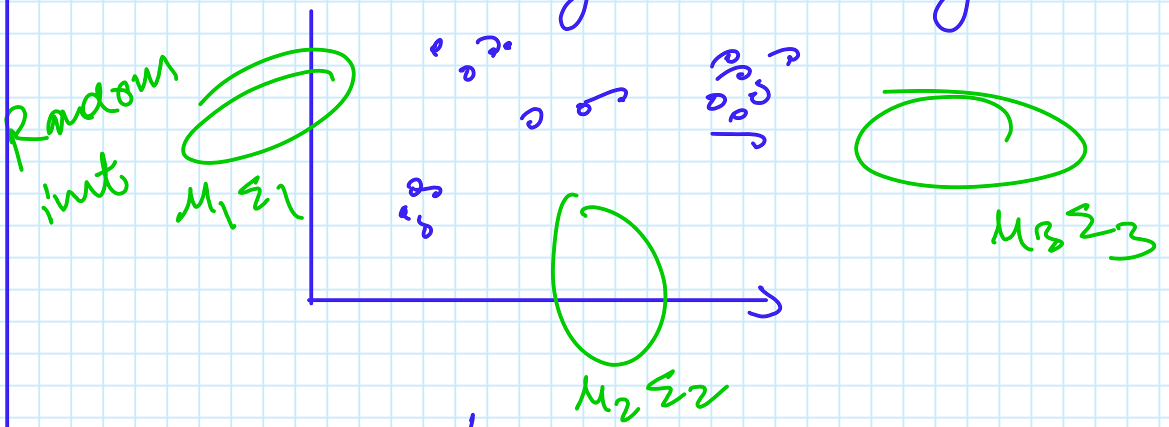
→ Assign related points to nearest centroid

→ Update the centroid $\left(\frac{\Sigma x}{n}\right)$

Expectation - maximisation algorithm

(k=3)

E-M → Randomly initialise gaussians (μ, Σ) for each



Step 2 → Assign/find prob of each pt belonging to each gaussians

⇒ for each x_i → 1 to N

$x_i \rightarrow \begin{cases} G_1 \rightarrow 0.2 \\ G_2 \rightarrow 0.3 \\ G_3 \rightarrow 0.4 \end{cases}$

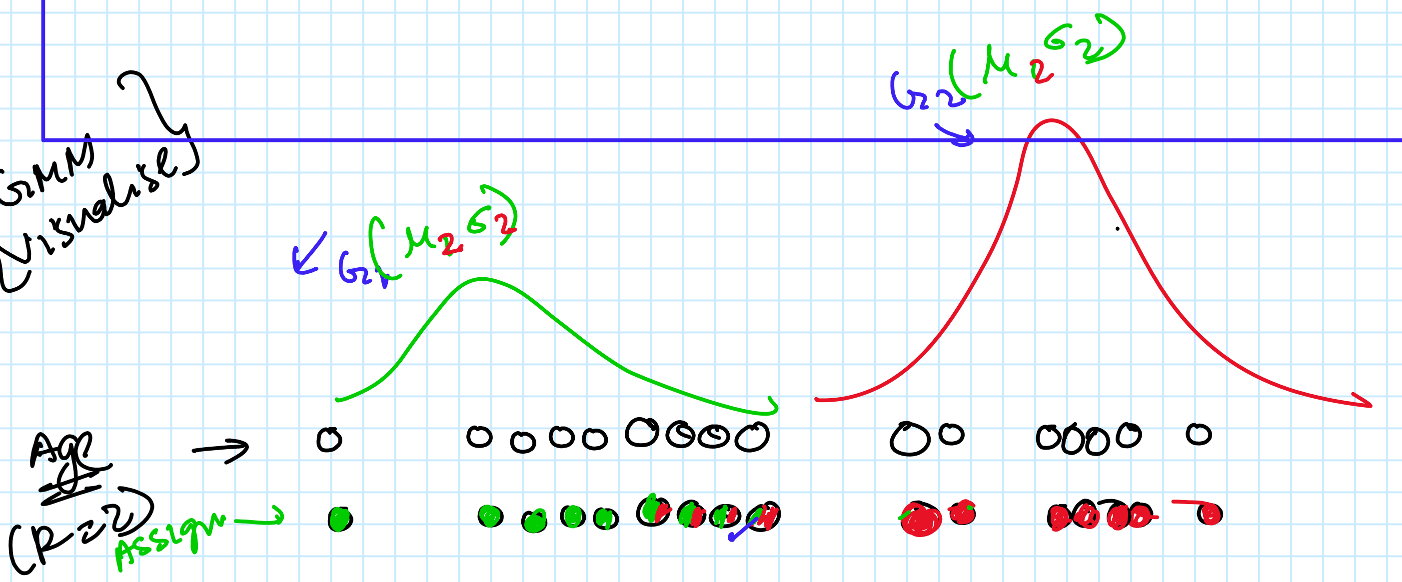
⇒ x_1 → G_1 20%, G_2 30%, G_3 50%

⋮

x_N

Step 3 Update the centroid (K Means) / → update the gaussians (GMM) → μ, Σ

(GMM Visualise)



Update → formula

K means → $\left(\frac{\Sigma x_i}{N}\right)$ $i \in C_1$

Update formula → weighted Average → $\frac{\Sigma w_i x_i}{N}$ / $\frac{\Sigma (p_i)(x_i)}{N}$ → (0 or 1) → K means → $\frac{\Sigma x_i}{N}$

GMM

15 February 2024 22:53

EM →

Random Init gaussians

$x_i \rightarrow C_1, C_2, C_3$

Step-1 Expectation:

- This is basically an assignment step.
- In this step, for each point x_i, we compute the probability of point x_i belonging to a j^{th} cluster
- So, initially we start by randomly assigning a probability such that it belings to j^{th} gaussian (j^{th} cluster).
- We then compute the probabilities of a point belonging to k different clusters using Probability Density Function

Step-2. Maximization:

- In GMMs, we compute normalized probabilities of each and every point belonging to a cluster.
- This is kind of like a soft assignment, whereas in KMeans we did the Hard Assignment by randomly picking centroids.
- In the Maximization step we re-estimate gaussian parameters, for all K gaussians.

$\mu_j = \frac{\sum P_i x_i}{N}$

K Means → $\frac{\sum x_i}{N}$ → In GMM → $\frac{\sum P_i \cdot x_i}{N}$

Very high Dimensional → Initial^n issue

GMM Algorithm & Implementation

There are two steps involved in GMM Algorithm:

1. Expectation
2. Maximization

Step-1. Expectation:

- This is basically an assignment step.
- In this step, for each point x_i, we compute the probability of point x_i belonging to a j_{th} cluster
- So, initially we start by randomly assigning a probability such that it belings to j_{th} gaussian (j_{th} cluster).
- We then compute the probabilities of a point belonging to k different clusters using Probability Density Function

Step-2. Maximization:

- In GMMs, we compute normalized probabilities of each and every point belonging to a cluster.
- This is kind of like a soft assignment, whereas in KMeans we did the Hard Assignment by randomly picking centroids.
- In the Maximization step, we re-estimate gaussian parameters, for all K gaussians.

Q. How do we do that?

- We update mean vectore (μ_j) and covariance vectors ($\sum_j$) using weighted scheme

. **What is weighted scheme?**

- Suppose we've to update mean vector(μ_j)
- Instead of taking simple average as we do in K-Means, we compute weighted average of each x_i, where the weightage is based on the probabilities that the point $x_i \in$ cluster j
- We also compute covariance vector in the same fashion
- In the seond step of K-means, we update the points based on the updated centroids of the cluster
- Whereas in GMM, we update a point based on the mean (which is same as centroid in K-Means) and covariance. The only difference is that we use weighting scheme in GMM Models